OIPE – IAP MAY 12 2006 PATENT & TRADEMARK OFFICE

Certification Under 37 CFR §1.10
Express Mail Label Number: EV 336 060 171 US Date of Deposit: May 12, 2006
I hereby certify that this transmittal and the documents and/or fees hereon and attached hereto have been deposited as "Express Mail Post Office to Addressee" in accordance with 37 CFR §1.10, with Express Mail Label Number listed above.

By: [illegible]
Susan L. Baka

PATENT

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Appl. No.	**:**	**10/005,729**	**Confirmation No.**	**:**	**9132**
Applicant	**:**	**Holler et al.**	**TC/A.U.**	**:**	**3621**
Filed	**:**	**November 6, 2001**			
Examiner	**:**	**Cristina O. Sherr**			
Docket No.	**:**	**30126-8016.US01**	**Customer No.**	**:**	**22918**

Declaration of Prior Invention Under 37 C.F.R. § 1.131

Commissioner for Patents
P.O. Box 1450
Alexandria, VA

I. This Declaration establishes invention prior to September 26, 2000.

II. This Declaration is being made by Anne Marie Holler, Lacky Vasant Shah, Sameer Panwar, and Amit Patel, i.e., the named inventors of the above-identified patent application.

III. Conception: Prior to September 26, 2000, we conceived the inventions currently presented in independent claims 1, 19, and 37, of the above-identified patent application. A list of these claims is attached hereto as Exhibit A. Claim 1 is exemplary of an embodiment of the inventions. Exhibit B includes a listing of files related to a product that is representative of the embodiment claimed in the exemplary independent claim 1. Exhibit B includes versions of software and documentation that were created in a Content Management System (CMS) prior to September 26, 2000. The dates of each file have been redacted.

Exhibit C includes a subset of content from the CMS as of September 25, 2000. The content is from files listed in Exhibit B, which are entitled Exhibit C1 through C18. Exhibit C correlates to the exemplary independent claim 1. These

correlations are for the purpose of example only, and not intended to limit the scope of the claims. TABLE 1 provides a rough correlation between Exhibit C and, for example, independent claim 1:

TABLE 1

EXHIBIT C (Examples only)	CLAIM 1
C2) • Functionality (pg. 1) o The Client Network Interface (CNI) provides the interface for sending message to servers and provides threads for receiving responses and dispatching them appropriately. (para. 1). • Asynchronous Server Calls (pgs. 5 – 6) o The network send thread is periodically awoken, and it coalesces requests off the NW request queue and sends them to the server (pg. 6, para. 1). o The network receives thread waits for responses to come back from any server. (pg. 6, para. 2). o Finally, the response dispatch thread pulls responses off the response queue, and handles the work of dispatching them appropriately. (pg. 6, para. 3) C6) • Estream client network interface o Handles requests from Estream cache manager o Handles protocol interface to/from server C7) Diagram illustrating structure of server streaming of application programs across a computer network while executing application programs on an Estream client. C17) Abstract and descriptions of CORBA illustrate an implementation of a server framework for Estream. (pg. 1). C18) Diagram illustrating structure of server streaming of applications programs across a computer network while executing application programs on an Estream client.	A process for the delivery of server-based streamed applications and data to a client and the management of said streamed applications on a server, the process comprising:

(pg. 7)	
C1) • **AIMInstallApplication Prototype (pg. 16)** o **Step 5. Initializing the profile and prefetch data for this app.** **C7) Estream client-server diagram comprising an application server. Diagram illustrating structure of server streaming of application programs across a computer network while executing application programs on an Estream client.** **C9)** • **Application stream set builder (pg. 12)** o **The application stream set is built on a computer.** **C10)** • **Functionality (pg. 1)** o **The AppInstallBlock is a block of code and data associated with a particular application. This AppInstallBlock contains the information needed by the Estream client to "initialize" the client machine before the Estream application is used for the first time.** **C12)** • **File system paradigm (pg. 12)** o **After the application is stored on the server…, the client prototype is started.** **C16)** • **The primary job of the App Server is to service client requests for application data (pg. 1)** o **Compressed application data is persistently saved.**	**providing application set storage means for persistently storing streamed application program sets on a server;**
C9) • **Technical description of the invention. (pg. 12)**	**wherein said streamed application sets contain streamed application file pages;**

○ An application file server: Responds to requests by client application cache manager for portions of application's files and directory structure on the server. Transmits compressed information for better bandwidth utilization. (pg. 12, item #4). C16) • Functionality (pg. 1) ○ The primary job of the App Server is to service client requests for application data blocks. (pg. 1, para. 4). ○ The App Server serves data derived from Estream Sets. (pg. 1, para. 5).	
C5) Estream Server Functionality • Application server is functionally read-only (pg. 2) C7) Estream client-server diagram illustrating transmission of Estream sets. C13) Estream Builder data flow diagram illustrating the conversion of locally installable applications into Estream sets. C16) • The Application Server is optimized to do one thing only: serve pages from the read-only file system of eStream (pg. 4) C18) • Application Server (pg. 26) ○ The application server is there to handle read requests for files accessed by Estream clients. Any file accessed on a client through the EFS can have this read request passed to an app server.	wherein said streamed application file pages are read only;
C7) Estream client-server diagram illustrating transmission of Estream sets to Estream client.	providing means for receiving client requests for streamed application file pages;

C9) An application file server responds to requests by client application cache manager for portions of application's files (pg. 12). **C16) Each server responds to a single self-contained message from a client requesting a page set from the server (pg. 1).** **C18)** • Application Server (pg. 26) o **The application server is there to handle read requests for files accessed by Estream clients. Any file accessed on a client through the EFS can have this read request passed to an app server.**	
C3) • ECM-LSM Interaction (pg. 9) o **When ECM receives a request, it cheks to determine if an access token is available.** **C5)** • Execution of application (pg. 2) o **Send unique certificate for application to appropriate ASP DRM Server, get back id for closest/best App Server & session id.** o **Contact designated App Server using id info, request file system data as necessary. Respond to running application's requests, collect usage data.** • DRM server (pg. 2) o **DRM server determines authentication of a client, validates application licenses and so on.** **C8)** • Accessing Files (pg. 3) o **The client would provide the file id of interest and the proper**	**providing validation means for validating whether a client has access privilege to a requested streamed application file page;**

access token for a file. o The system verifies it. C9) • The execution controller is given an argument indicating which application is to be executed. From the point of view of the client and its operating system, the application is resident locally on the client; the execution controller negotiates with an appropriate server to allow the client to obtain (as needed) segments of the associated application files located on the servers. (pg. 12, para. 2). • If a server accepts the task of serving the application to the client, the execution controller passes the application access request on to the application remote file interface code. This code allows the client to reference file and directory information associated with the remote application as if it resided on a local physical disk device. (pg. 13, 3rd full paragraph). C16) • Security (pg. 9) o There are two levels of security involved in the AS. One of them is to require the client to have an AccessToken and presents it to the AS upon every request.	
C3) • Functionality (pg. 1) o The cache manager manages the on-disk cache of file system data, and the in-memory data structures for managing this cache. (pg. 1, para. 3). • Diagram illustrating overall client architecture comprising cache	providing caching means for storing commonly accessed streamed application file pages in a cache;

elements. (pg. 8). • Implementation of the cache manager. (pgs. 11 – 17). C4) • Cache organization (pg. 1) o The cache will be contained in 2 or more files. One file will contain the cache indices, and one or more files will contain the data blocks for cached files. C9) • The client's operating system begins executing the requested application located remotely on a server. The operating system memory-maps the application and begins executing it, with the application remote file interface code obtaining control whenever the client system's page fault handler determines that the application's page is located on the remote disk drive. The page fault handler asks the application remote file interface code to place the appropriate page data in main memory. The application remote file interface code sends a request to the cache manger for the desired data. (pg. 13, 4th full paragraph). C12) • Data flow description (pg. 6) o The OS loads the application executable into memory and runs the executable. (pg. 7, step 12). o The executable file image is loaded into memory and starts executing. The application files will continuously be loaded into memory as needed. (pg. 7, step 13).	
C3) • The cache manager (pg. 1)	wherein said requested streamed application file page is retrieved from said caching

o **The cache manager manages cache of file system data by reading information from the cache.** o **It does not manage prefetching of data from the server. That is the role of the eStream Profiling and Fetching (EPF) component** C5) • Execution of application (pg. 2) o **Contact designated App Server using id info, request file system data as necessary. Respond to running application's requests, collect usage data. Cache portions of applications, file system info, & user preference info.**	**means if it is resident in said cache, otherwise said requested streamed application file page is retrieved from said application set storage means;**
C5) • Execution of application (pg. 2) o **Contact designated App Server using id info, request file system data as necessary. Respond to running application's requests, collect usage data. Cache portions of applications, file system info, & user preference info.** C16) • App Server (pg. 1) o **The App Server serves client requests for application data blocks.** o **The App Server is designed to minimize the amount of CPU time to satisfy each client request.**	**wherein clients request streamed application file pages using a unique set of numbers common among all servers that store the particular streamed application file pages; and**
C3) • File system paradigm (pg. 12) o **After the application is stored on the server…, the client prototype is started.**	**providing means for sending said requested streamed application file page to said client.**

C5) • **Execution of application (pg. 6)** o **Contact designated App Server using id info, request file system data as necessary. Respond to running application's requests, collect usage data. Cache portions of applications, file system info, & user preference info.** **C7) Estream client-server diagram comprising an application server.**	

IV. <u>Diligence:</u> We diligently constructively reduced the invention to practice on Nov. 6, 2001. Attached, with dates redacted, as Exhibits D1 through D5 (collectively "Exhibit D") are exemplary documents produced between September 26, 2000 and constructive reduction to practice. These documents are in chronological order, and have redacted dates which occurred at irregular intervals but without interruption extending from our conception of the invention to our constructive reduction to practice of the invention. Exhibit D is as follow:

a) D1: Estream 1.0 planning document

b) D2: Estream server component framework low level design

c) D3: Estream set format low level design

d) D4: Estream 1.0 high level design

e) D5: Estream web server load monitoring applet low level design

Exhibit D correlates to the exemplary independent claim 1. These correlations are for the purpose of example only, and not intended to limit the scope of the claims. TABLE 2 provides a rough correlation between Exhibit D and, for example, independent claim 1:

TABLE 2

EXHIBIT D (Examples only)	**CLAIM 1**
D1) Server group time estimates for implementation. **D2)** • **Functionality (pg. 1)** o **The Server Component Framework provides a common basis on which server components are implemented. The framework provides a number of services such as common server initialization and configuration, messaging, state management, logging, and error handling. (pg. 1, para. 1).** **D3)** • **Functionality (pg. 1)** o **The Estream set is a data set associated with an application suitable for streaming over the network.** **D4)** • **The eStream builder (pg. 2)** o **It is used to create an eStream set for the application.** **D5)** • **Functionality (pg. 1)** o **One of the requirements for the Estream web server is a facility for monitoring server load. Per this document, this facility will be provided by a graphical load-monitoring applet that will be available for deployment at customer sites as part of the Estream web server installation.**	**A process for the delivery of server-based streamed applications and data to a client and the management of said streamed applications on a server, the process comprising:**
D2) • **Functionality (pg. 1)** o **The Server Component Framework provides a common basis on which server components are implemented. The framework**	**providing application set storage means for persistently storing streamed application program sets on a server;**

provides a number of services such as common server initialization and configuration, messaging, state management, logging, and error handling. (pg. 1, para. 1). • All configuration information is persistently stored within the database (pg. 4).	
D1) • Server group o App Server time estimate for implementation. D3) • Functionality (pg. 1) o The Estream set is a data set associated with an application suitable for streaming over the network. D4) • Server components (pg. 10) o App Server is essentially a file server for eStream sets. It satisfies requests for "pages" from eStream files from the client. D5) • Functionality (pg. 1) o One of the requirements for the Estream web server is a facility for monitoring server load. Per this document, this facility will be provided by a graphical load-monitoring applet that will be available for deployment at customer sites as part of the Estream web server installation.	wherein said streamed application sets contain streamed application file pages;
D3) • EStream directory (pgs. 3-4) o Metadata consists of file byte size and Bit 0 is read-only set if a file is read-only. o Bit 2 of eStream flags is read-only	wherein said streamed application file pages are read only;

if a file is read-only.	
D4) • **App server** ○ **The application server is there to handle read requests for files accessed by Estream clients. (pg. 10).**	**providing means for receiving client requests for streamed application file pages;**
D3) • **EStream Directory (pgs. 3-4)** ○ **The bits of the eStream flags have the following meaning: Bit 1 reqiures AccessToken if a file requires access token before the client can read it.** **D4)** • **Overview (pgs. 1-2)** ○ **Authentication using tokens supplied by a license server to each active client is performed.** ○ **Before any page request is fulfilled, the client license subscription manager (LSM) will check that the user has permission to run the application, requesting an access token from the server.** ○ **This valid access token is sent from the client to the server for every page request; this authenticates the request.** • **ECM (pgs. 8-9)** ○ **The ECM works with the LSM to insure that all applications have appropriately validated licenses before their files are accessed.** • **Slim server (pg. 10)** ○ **Slim server handles requests from a client for user and service provider information and grants access tokens to the client for executing eStream applications.**	**providing validation means for validating whether a client has access privilege to a requested streamed application file page;**
D4)	**providing caching means for storing**

• Overview (pg. 1) o A distributed file system for application files, residing on a server and cached on a client. • Client components (pg. 7) o ECM: the Estream cache manager. This is the user-space component that handles requests from the Estream File System Driver, and manages the on-disk and in-memory cache of file contents.	commonly accessed streamed application file pages in a cache;
D1) • Client group o Estream cache manager time estimate for implementation. D4) • Overview (pgs. 1-2) o Client requests will be forwarded from the EFSD to the eStream cache manager (ECM), and on to the app server, assuming the page requested is not in the cache. • Client components (pgs. 7-8) o ECM: the Estream cache manager. This is the user-space component that handles requests from the Estream File System Driver, and manages the on-disk and in-memory cache of file contents. o ECM handles all file requests from the EFSD, either by using previously cached contents or requesting the contents from a server.	wherein said requested streamed application file page is retrieved from said caching means if it is resident in said cache, otherwise said requested streamed application file page is retrieved from said application set storage means;
D2) • A common set of configurable parameters is defined for all server components (pg. 4) D4)	wherein clients request streamed application file pages using a unique set of numbers common among all servers that store the particular streamed application file pages; and

• **Client components (pg. 7)** o **ECM: the Estream cache manager. This is the user-space component that handles requests from the Estream File System Driver, and manages the on-disk and in-memory cache of file contents.** • **Overview** o **A small client "player" program to allow local execution of applications that reside on the servers. (pg. 1)** o **The user will now see standard shortcuts for subscribed applications, exactly as though the app were installed locally. (pg. 2, 6th box)**	
D4) • **App server (pg. 10)** o **This will respond to both synchronous (demand fetching) and asynchronous (prefetching) page requests from many different clients, for many different types of applications and files within those applications.**	**providing means for sending said requested streamed application file page to said client.**

V. **We hereby declare that all statements made herein of our own knowledge are true and that all statements made on information and belief are believed to be true; and further that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, (18 U.S.C. §1001) and that such willful false statements may jeopardize the validity of this application or any patent issued thereon.**

Anne Marie Holler

Date

Lacky Vasant Shah

Date

Sameer Panwar

Date

Amit Patel

Amit Patel

Date 5/4/2006